

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPNET** (**UPGRADE European Network**).

<<http://www.ati.es/novatica/>>
<<http://www.ati.es/reicis/>>
<<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC**, **RITSI** e **Hispalinux**, junto a la que participa en **Prolinnova**.

Consejo Editorial

Juan Batlle Montserrat, Rafael Fernández Calvo, Luis Fernández Sanz, Javier López Muñoz, Alberto Lobet Ballfiori, Gabriel Martí Fuentes, Josep Molas i Bertran, José Onofre Montesca Andrés, Olga Pallas Ovejún, Fernando Píera Gómez (Presidente del Consejo), Ramon Puigjaner Trepal, Miquel Sarries Grifó, Adolfo Vázquez Rodríguez, Asunción Yurbe Herranz

Coordinación Editorial

Llorenç Pagés Casas <lpages@ati.es>

Composición y autoedición

Jorge Llácer Gil de Ramales

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>> Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores

Acceso y recuperación de la información

Asa María Gómez Hidalgo (Optinet), <imgomez@yaho.es>

Manuel J. María López (Universidad de Huelva), <manuel.marja@diesta.uhu.es>

Administración Pública electrónica

Francisco López Crespo (MAE), <flc@ati.es>

Arquitecturas

Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>

Jordi Tubella Morgadas (DAC-UPC), <jordit@ac.upc.es>

Auditoría TIC

Marina Touriño Troilo, <marinatourino@marinatourino.com>

Manuel Palao García-Suelto (ASIA), <manuel@palao.com>

Boracho y tecnologías

Isabel Hernández Collazos (Fac. Derecho de Donostia, UPV), <ihernando@legalek.net>

Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

Excedencia Universitaria de la Informática

Cristóbal Pareja Flores (DSIP-UCM), <cpareja@si.ucm.es>

J. Angel Velázquez Hurtado (ESCET-URJC), <a.velazquez@esctet.urjc.es>

Entorno digital personal

Alonso Álvarez García (TID), <aag@tid.es>

Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>

Estándares Web

Encarnación Quesada Ruiz (Oficina Española del W3C) <equesada@w3.org>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería) <jcarco@gmail.com>

Gestión del Conocimiento

Juan Ramón Solé (Cap Gemini Ernst & Young), <jra.sol@ati.es>

Informática y Filosofía

José Angel Olivás Varela (Escuela Superior de Informática, UCLM) <joseangel.olivas@uclm.es>

Karim Gherab Martin (Harvard University) <kgherab@gmail.com>

Informáticas Gráficas

Miguel Chover Sellés (Universitat Jaume I de Castellón), <mchover@si.uji.es>

Roberto Vivó Hernández (Eurographics, sección española), <rvivo@dstc.upv.es>

Ingeniería del Software

Javier Dolado Cosín (CIS-UPV), <dolado@si.ehu.es>

Luis Fernández Sanz (PRIS-El-UEM), <lufern@dpri.es.uem.es>

Inteligencia Artificial

Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV)

<vbotti.vmj@dsic.upv.es>

Interacción Persona-Computador

Julio Abascal González (FI-UPV), <julio@si.ehu.es>

Lenguaje e Informática

M. del Carmen Ugarte García (IBM), <cuarte@ati.es>

Lenguajes Informáticos

Andrés Marín López (Univ. Carlos III), <amarin@it.uc3m.es>

Oscar Belmonte Fernández (Univ. Jaime I de Castellón), <belfern@lsi.uji.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo), <xggo@uvigo.es>

Manuel Patomar (Univ. de Alicante), <mpatomar@dsi.ua.es>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI) <gnu.fede@gmail.com>

Mikel Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid), <mikelxni_uni@yahoo.es>

Profesión Informática

Rafael Fernández Calvo (ATI), <rfoval@ati.es>

Miquel Sarries Grifó (Ayto. de Barcelona), <msarries@ati.es>

Redes y servicios telemáticos

José Luis Marzo Lázaro (Univ. de Girona), <jlmarzo.marzo@udg.es>

Germán Santos Sosa (UPC), <german@ac.upc.es>

Seguridad

Javier Arellano Bertolin (Univ. de Deusto), <jarellito@eside.deusto.es>

Javier López Muñoz (ETS Informática-UMA), <jlm@icc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <alalmonso.puente@dit.upm.es>

Software Libre

Jesus M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC), <jmgh.pheras@gsyc.esctet.urjc.es>

Tecnología de Bibliotecas

Jesus Garcia Molina (DS-UM), <jmolina@um.es>

Gustavo Rossi (LIFIA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>

Tecnologías para la Educación

Juan Manuel Doderio Berardo (UC3M), <doderio@inf.uc3m.es>

César Pablo Córcoles Briondo (UOC), <ccorcoles@uoc.edu>

Tecnologías y Empresa

Didac López Vilas (Universitat de Girona), <didac.lopez@ati.es>

Francisco Javier Gamais Sánchez (Indra Sistemas), <jfgamais@gmail.com>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <aguayo.guevara@icc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º, dcha., 28006 Madrid

Tf: 91 402 93 91; fax: 91 309 36 85 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Remo de Valencia 23, 46105 Valencia

Tf: /fax 96 33 30 392 <secreal@ati.es>

Administración y Redacción ATI Cataluña

Via Lestatina 45, ppal. T: 08003 Barcelona

Tf: 93 41 25 235; fax: 93 41 27 713 <secregen@ati.es>

Redacción ATI Andalucía

Isaac Newton, s/n, Ed. Sadele,

Isa Cortiús, 41092 Sevilla. Tf: /fax 95 44 60 779 <secreand@ati.es>

Redacción ATI Aragón

Lagascá 9, 3-B, 50006 Zaragoza.

Tf: /fax 97 62 35 181 <secrear@ati.es>

Redacción ATI Asturias-Cantabria

<gp-astucant@ati.es>

Redacción ATI Castilla-La Mancha

<gp-clmancha@ati.es>

Suscripción y Ventas <<http://www.ati.es/novatica/interes.html>>, ATI Cataluña, ATI Madrid

Publicidad

Padilla 66, 3º, dcha., 28006 Madrid

Tf: 91 402 93 91; fax: 91 309 36 85 <novatica@ati.es>

Impresión: Dierra S.A., Juan de Austria 66, 08005 Barcelona

Depósito legal: B 15.154-1975 - ISSN: 0211-2124. CODEN NOVAEC

Partida: "Razón aurea" - Concha Arias Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

editorial

La elaboración de normas en el ámbito de las TIC

> 02

en resumen

El corazón del problema

> 02

Llorenç Pagés Casas

IFIP

Reunión del TC1 (Foundations of Computer Science)

> 03

Michael Hinchey, Joaquim Gabarró Vallés

Task Force de IFIP para estudiar el nuevo modelo de afiliación

> 03

Ramon Puigjaner Trepal

monografía

Desarrollo de Software Dirigido por Modelos

(En colaboración con **UPGRADE**)

Editores invitados: Jean Bézivin, Antonio Vallecillo Moreno, Jesús García Molina y Gustavo Rossi

Presentación. Siete años de MDA®: pasado, presente y futuro

> 04

Jean Bézivin, Antonio Vallecillo Moreno, Jesús García Molina, Gustavo Rossi

Una breve historia de MDA

> 09

Andrew Watson

Manifestaciones sobre MDA

> 13

Bran Selic

Entornos de desarrollo integrados específicos de dominio

> 17

Steve Cook, Stuart Kent

Inteligencia de modelos: un enfoque para guiar el modelado

> 21

Jules White, Douglas C. Schmidt, Andrey Nechypurenko, Egon Wuchner

Diferencias entre modelos en Eclipse EMF

> 28

Cédric Brun, Alfonso Pierantonio

Arquitectura dirigida por modelos en Eclipse

> 33

Richard C. Gronback, Ed Merks

Ingeniería Web dirigida por modelos

> 37

Nora Koch, Santiago Meliá Beigbeder, Nathalie Moreno Vergara,

Vicente Pelechano Ferragud, Fernando Sánchez Figueroa, Juan Manuel Vara Mesa

secciones técnicas

Arquitecturas

Microbots: Fundamentos y aplicaciones

> 42

José María Rodríguez Corral, Arturo Morgado Estévez, Francisco Cordón González,

Rafael González Chacón, Ignacio García Vargas

Eenseñanza Universitaria de la Informática

Preguntas frecuentes y nuestras respuestas favoritas sobre

la pertinencia de los métodos docentes centrados en el estudiante para

adaptar una asignatura al EEES

> 48

Miguel Valero-García, Juan José Navarro Guerrero

Estándares Web

DENEB: Una plataforma para el desarrollo y ejecución de procesos Web dinámicos

> 51

Javier Fabra Caro, Pedro Álvarez Pérez-Aradores, José Angel Bañares Bañares, Joaquín Ezpeleta Mateo

Redes y servicios telemáticos

Una radiografía del estado de Internet

> 56

José Luis Gahete Díaz, Natalia Fernández Gallego, Ana María Garzo Ortega, Gonzalo Martín Villaescusa

Referencias autorizadas

> 62

sociedad de la información

Comunidades y trabajo colaborativo

Prácticas científicas y Sociedad del Conocimiento:

el ejemplo de las comunidades FLOSS

> 68

Roberto Feltrero Oreja

Programar es crear

Voto Útil (CUPCAM 2007, problema B, enunciado)

> 73

Manuel Freire Morán, Julio Mariño Carballo

Buscando orejas (CUPCAM 2007, problema C, enunciado)

> 74

Dolores Lodares González

Polígonos Escalera (CUPCAM 2007, problema A, solución)

> 75

Manuel Abellanas Oar, Manuel Freire Morán

asuntos interiores

Coordinación Editorial / Programación de Novática / Socios Institucionales

> 77

Monografía del próximo número: "El futuro de la tecnología educativa"

Javier Fabra Caro, Pedro Álvarez Pérez-Aradros, José Ángel Bañares Bañares, Joaquín Ezpeleta Mateo

Instituto de Investigación en Ingeniería de Aragón, Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza

{jfabra, alvaper, banares, ezpeleta}@unizar.es

1. Introducción

El éxito de Internet ha ofrecido a las organizaciones una ocasión única de encontrar nuevas oportunidades de negocio a través de la Red. Para poder participar en este nuevo mercado, las organizaciones exponen sus procesos de negocio como *procesos Web*.

En la actualidad, numerosos trabajos de investigación apuestan por separar explícitamente la lógica de negocio de los procesos (sus *workflows*) de la lógica de interacción (sus *protocolos de coordinación*) y sus respectivas *conversaciones*). En general, estos trabajos reutilizan los estándares de servicios Web para la especificación de ambas lógicas (por ejemplo, BPEL4WS para la lógica de negocio y WSCI, WS-CDL o OWL-S para la lógica de interacción). Esta decisión provoca que las respectivas plataformas de ejecución de procesos sean construidas integrando las herramientas que existen en el mercado en torno a estos estándares. Este modelo de solución no está exento de problemas [1], originados, fundamentalmente, por la falta de una visión global e integrada de los estándares de especificación, las diferencias en cuanto al grado de madurez de los estándares involucrados y, finalmente, la falta de metodologías que establezcan cómo construir un sistema complejo a partir de estas especificaciones. Como resultado, la mayoría de las plataformas de ejecución de procesos propuestas son prototipos funcionalmente muy simples [1][2][3] o complejos diseños arquitecturales pendientes de ser implementados.

Todas estas propuestas asumen un modelo estático de procesos Web en el cual tanto la lógica de negocio como la lógica de interacción de los procesos se define en tiempo de diseño. Por lo tanto, en el momento en el que se inicia la ejecución de un proceso, la estructura de su *workflow* está preestablecida (qué actividades deben completarse y en qué orden) y se ha decidido con qué procesos externos va a cooperar y en base a qué protocolos de interacción. En este tipo de escenarios, cada organización determina de forma independiente la lógica de negocio de sus procesos. Sin embargo, desde el punto de vista de las interacciones, las organizaciones deben decidir conjuntamente qué protocolo

DENEB: Una plataforma para el desarrollo y ejecución de procesos Web dinámicos

Este artículo fue seleccionado para su publicación en **Novática** entre las ponencias presentadas a las III Jornadas Científico-Técnicas en Servicios Web y SOA (JSWEB2007) celebradas en Zaragoza y de las que ATI fue entidad colaboradora.

Resumen: *la naturaleza dinámica de los entornos inter-organizacionales exige nuevos modelos de procesos de negocio y plataformas tecnológicas para su desarrollo y ejecución. La mayoría de los trabajos de investigación han apostado por reutilizar los estándares de servicios Web y las tecnologías Web para la especificación de los procesos y la implementación de estas plataformas orientadas a procesos, respectivamente. Aunque a priori esta decisión está suficientemente justificada, los resultados obtenidos no han sido los deseados. En este artículo se describe el diseño y la implementación de la plataforma DENEB de desarrollo y ejecución de procesos inter-organizacionales que resuelve las limitaciones de las actuales soluciones basadas en estándares. Funcionalmente, los procesos en ejecución pueden configurar dinámicamente su lógica de negocio y su lógica de interacción, así como interoperar con otros procesos remotos y heterogéneos. La clave está en la utilización conjunta del paradigma de Redes-en-Redes y la herramienta Renew.*

Palabras clave: BPM, procesos Web, protocolos de interacción, redes de Petri.

los estándar de interacción utilizan para cooperar o deben consensuar nuevos protocolos no estándar.

Por desgracia, este modelo de procesos es demasiado rígido para ser directamente aplicado en entornos inter-organizacionales, caracterizados por una continua fluctuación de los procesos disponibles, cambios frecuentes en las políticas y estrategias de negocio, fallos y condiciones de ejecución variables [10]. Por lo tanto, deben proponerse nuevos modelos de procesos que sean más flexibles y permitan configurar en tiempo de ejecución la lógica de negocio y la lógica de interacción de los procesos (es decir, modelos de procesos Web dinámicos) y desarrollarse plataformas software para la ejecución de los procesos correspondientes.

En este artículo se presenta DENEB (del inglés, *platform for the Development and Execution of iNteroperable dynamic wEB processes*), una evolución de la plataforma descrita en [4][5] y basada en el paradigma de Redes-en-Redes [6], el modelo de coordinación Linda y la herramienta Renew [7][8]. DENEB es independiente de tecnologías concretas y, desde un punto de vista funcional de alto nivel, es capaz de ejecutar los *workflows* de los procesos, interpretar dinámicamente las conversaciones necesarias para interactuar con otros procesos e interoperar con otras plataformas remotas y heterogéneas (por ejemplo, con plataformas construidas utilizando los estándares de servicios Web). La propuesta utiliza las Redes-en-Redes como único formalismo para descri-

bir la lógica de negocio de los procesos Web, sus protocolos de coordinación e, incluso, la propia implementación de la plataforma de ejecución [11]. Esta decisión facilita la integración natural de los aspectos de composición y coordinación de los procesos.

2. Procesos Web dinámicos

A continuación se profundiza en las características que diferencian a los procesos Web definidos en tiempo de ejecución respecto a los definidos en tiempo de diseño. Fundamentalmente, los procesos Web dinámicos deben ser capaces de:

■ *Configurar en tiempo de ejecución su lógica de interacción y su lógica de negocio.* Durante su ejecución, un proceso dinámico debe descubrir los procesos con los que potencialmente podría colaborar y seleccionar el óptimo para la consecución de cada uno de sus objetivos. En cada momento, la cooperación con el proceso seleccionado puede estar definida por protocolos de coordinación previamente establecidos (por ejemplo, basados en estándares de interoperabilidad) o bien estar guiado por protocolos desconocidos (este último caso sería el escenario más general). Por otro lado, desde el punto de vista de la lógica de negocio, el orden en el que se ejecutan las actividades o tareas modeladas en el *workflow* del proceso no sólo depende de las estructuras de control, sino también del estado del entorno de ejecución.

■ *Monitorizar y reaccionar ante eventos esperados y/o inesperados.* Principalmente, estos eventos pueden ser fallos a nivel de sistema (por ejemplo, un proveedor que no

responde), fallos a nivel de lógica de negocio (el tiempo de respuesta de un proveedor ha sido mayor del esperado) o determinados estados de variables y/o entidades del entorno que son monitorizadas (un proveedor aplica descuentos temporalmente sobre sus productos). En cualquier caso, la detección de un evento implica la reconfiguración de la lógica de negocio y/o la lógica de interacción del proceso en base a políticas previamente establecidas.

■ *Interpretar y tomar decisiones en base a políticas de negocio.* Las políticas definen un conjunto de reglas que dotan a los procesos de cierta capacidad de "autogestión" en tiempo de ejecución. Por ejemplo, para configurar sus lógicas, recuperarse ante fallos, optimizar su funcionamiento, etc. Fundamentalmente, se distinguen dos tipos de políticas: a corto plazo (por ejemplo, durante la selección de un servicio, un proceso puede dar prioridad a aquellos proveedores que ofrezcan un tiempo de respuesta más eficiente) y a largo plazo (puede que ciertos proveedores, a priori eficaces, no hayan sido demasiado fiables en el pasado y el proceso prefiera evitar la colaboración con ellos).

El análisis de estos requisitos permite concluir que la capacidad de "configuración" en tiempo de ejecución es la propiedad clave que deben satisfacer los procesos Web para ser dinámicos (nótese que el manejo de eventos, la recuperación de fallos, la optimización y la adaptación a nuevas políticas de negocio reutilizan la configuración para adaptar los procesos Web a las condiciones cambiantes del entorno de ejecución). Esta propiedad requiere la integración de nuevos componentes funcionales en las plataformas de ejecución de procesos y la definición

de nuevas abstracciones a nivel de modelado. En las siguientes secciones se describe cómo estos requisitos han sido integrados en DENEb.

3. Modelo arquitectural para plataformas de procesos Web

La **figura 1** muestra los principales componentes de una plataforma capaz de ejecutar procesos Web e interpretar sus conversaciones. Este modelo arquitectural es una instancia de la arquitectura SOA (*Service Oriented Architecture*) y es válido independientemente de que sus procesos sean definidos en tiempo de diseño o de ejecución.

El **componente de composición** integra un intérprete de *workflows* para la ejecución de la lógica de negocio de los procesos. Esta lógica puede ser descrita utilizando un lenguaje de descripción de *workflows*, por ejemplo, BPEL4WS (*Business Process Execution Language for Web Services*). Básicamente, este tipo de lenguajes permiten definir qué actividades deben ejecutarse como parte del proceso y el orden en el que deben ser ejecutadas. Una actividad concreta puede ser interna al proceso (por ejemplo, una toma de decisión para determinar cuál es la siguiente actividad que va a ser ejecutada) o externa (por ejemplo, una invocación a una operación ofrecida a través de su interfaz por otro proceso externo). La ejecución de actividades externas es una forma de integrar en un proceso la funcionalidad ofrecida por otros.

Las interacciones entre los procesos no se limitan a la invocación de operaciones simples e independientes, sino que están definidas en base a *protocolos de coordinación* y sus respectivas *conversaciones*. Estos proto-

colos pueden ser descritos utilizando alguno de los lenguajes declarativos propuestos para servicios Web, por ejemplo, WSCI, WS-CDL o OWL-S.

En el modelo arquitectural propuesto, la gestión de las conversaciones se mantiene separada de la lógica de negocio. Esta separación facilita que un proceso en tiempo de ejecución pueda decidir con qué proceso externo desea conversar para completar una actividad e interpretar de forma dinámica el protocolo que defina esta interacción. El **intérprete de conversaciones** es el componente responsable de ejecutar la parte de la conversación que un proceso juega (conocida como *rol*) cuando interactúa con otros procesos.

Durante la ejecución de una conversación concreta, los procesos envían y reciben mensajes entre sí. Estos mensajes pueden ser intercambiados utilizando diferentes protocolos de comunicación (HTTP, SOAP, SMTP, RPC, JMS, etc.) y/o formatos de codificación (XML, binario, etc.). El objetivo del componente **broker de mensajes** es establecer una separación explícita entre la lógica de los mensajes (es decir, las conversaciones) y los mecanismos de comunicación y/o formatos de codificación concretos a través de los cuales se intercambian. Internamente, este componente consta de un *repositorio de mensajes*, donde se almacenan temporalmente los mensajes recibidos o pendientes de ser enviados, y de un conjunto de **mediadores** (en inglés *binding components*), que permiten la comunicación con procesos externos en base a protocolos de comunicación y formatos de codificación concretos. Nótese que estos mediadores permiten aislar la plataforma de tecnologías concretas

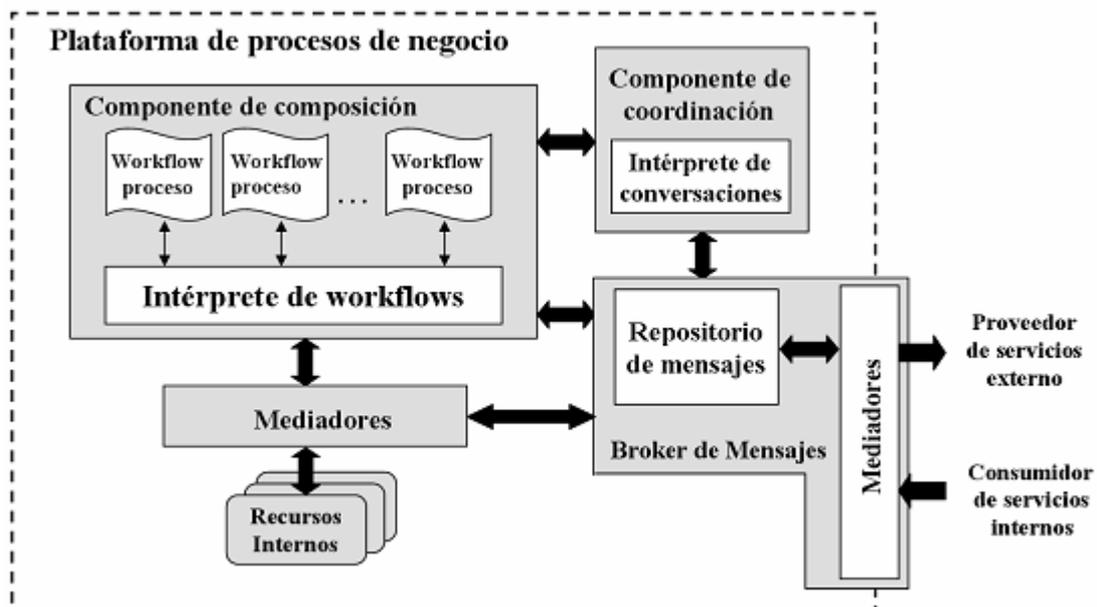


Figura 1. Modelo arquitectural de plataforma.

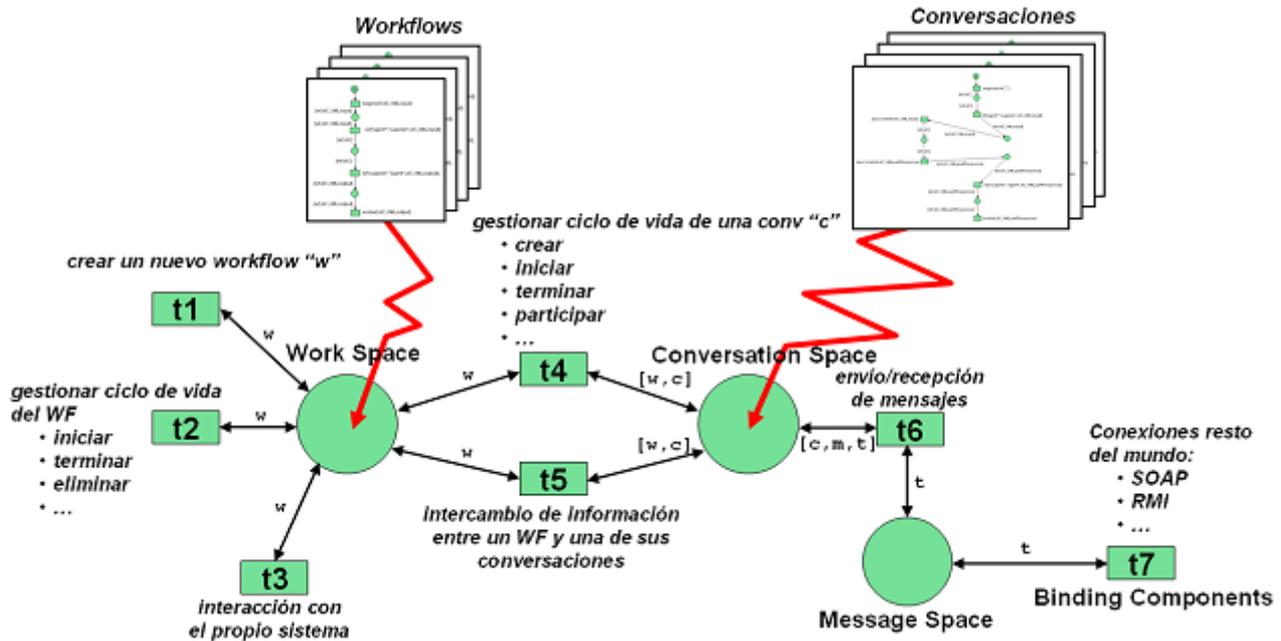


Figura 2. Vista abstracta de la implementación de la plataforma DENEb.

de comunicación e incluso de formatos de intercambio de información.

4. Diseño e implementación de la plataforma

El modelo arquitectural descrito en la sección 3 ha definido la implementación de la plataforma DENEb. Esta implementación está basada, fundamentalmente, en el paradigma de Redes-en-Redes y en la herramienta Renew.

La propuesta de implementación se basa en los siguientes aspectos:

- La concurrencia es un elemento clave en la composición y la coordinación de procesos. Las Redes de Petri (RdP) son una herramienta útil para abordar estos requisitos de concurrencia.
- Las interacciones entre los procesos tienen una naturaleza inherentemente asíncrona y están organizadas en conversaciones. La combinación de patrones de interacción simples (MEPs, *Message Exchange Patterns*) permite construir conversaciones y, a partir de éstas, protocolos.
- Además de almacenar, enviar y/o recibir mensajes, el componente *broker de mensajes* tiene que ser capaz de bloquear la ejecución de un proceso hasta que éste reciba un mensaje concreto. Como lenguaje intermedio para modelar las conversaciones entre los procesos se ha elegido el lenguaje de coordinación Linda [7] y una implementación distribuida de Linda, llamada *DRLinda* [9], como repositorio de mensajes.

En la figura 2 se muestra un esquema de alto nivel de la implementación de DENEb se-

gún las consideraciones de diseño anteriores (una descripción más detallada de la misma puede consultarse en [4] y [5]). Este esquema es una abstracción de los principales lugares y transiciones de la red (llamada *red del sistema* en terminología de Redes-en-Redes) que corresponde con la plataforma implementada.

Los *workflows* de los procesos y sus conversaciones son representados también como redes (llamadas *redes objeto* en terminología de Redes-en-Redes) y están contenidas en los lugares *Work-Space* y *Conversation-Space*, respectivamente.

A continuación se analiza el propósito de las distintas transiciones de la plataforma:

- Las transiciones *t1* y *t2* permiten crear un nuevo proceso (por iniciativa propia o en respuesta a una petición externa realizada por otro proceso) y gestionar el ciclo de vida de los procesos ya creados y que están contenidos en el *Work-Space*, respectivamente.
- La transición *t3* facilitan las interacciones de los procesos en ejecución con los recursos internos de la plataforma (por ejemplo, con una base de datos, un sistema experto o un sistema legado).
- Como se describió previamente, los procesos colaboran por medio de conversaciones. En una conversación concreta, cada proceso participante juega un determinado rol. La transición *t4* permite a un proceso crear una nueva conversación desempeñando un determinado rol e invocar a otros procesos que han declarado (en un registro UDDI, por ejemplo) aceptar jugar los roles complementarios (estos procesos podrán

estar en ejecución en la misma plataforma o en otra remota). También permite gestionar el ciclo de vida de las conversaciones previamente creadas. Por otro lado, la transición *t5* facilita las interacciones entre los procesos y sus conversaciones (por ejemplo, el *workflow* del proceso puede ser el responsable de tomar una decisión que afecte al desarrollo de la conversación).

■ Las conversaciones contenidas en el *Conversation-Space* constan de distintos MEPs que agrupan lógicamente mensajes. La transición *t6* encapsula las primitivas de comunicación Linda *write*, *take* y *read* (también conocidas en el modelo original como *out*, *in* y *rd* respectivamente) utilizadas en el envío y la recepción de los mensajes. Estas primitivas permiten el intercambio de mensajes con otros procesos externos a través del repositorio del *broker de mensajes* (representado en la figura 2 como el lugar *Message-Space*). Este componente ha sido implementado como un sistema de coordinación Linda modelado también en base al paradigma de Redes-en-Redes y ejecutado por la propia herramienta Renew (*DRLinda* [9]).

■ Finalmente, los mensajes intercambiados entre los procesos a través del *Message-Space* son representados internamente en un "lenguaje intermedio", más concretamente, como tuplas de Linda. Estas tuplas son independientes de la tecnología de interacción (es decir, de los protocolos de transporte y de los formatos de codificación) utilizada para la comunicación con los procesos externos. Los mediadores (o *binding components*) encapsulados en la transición *t7* son los responsables de esta transforma-

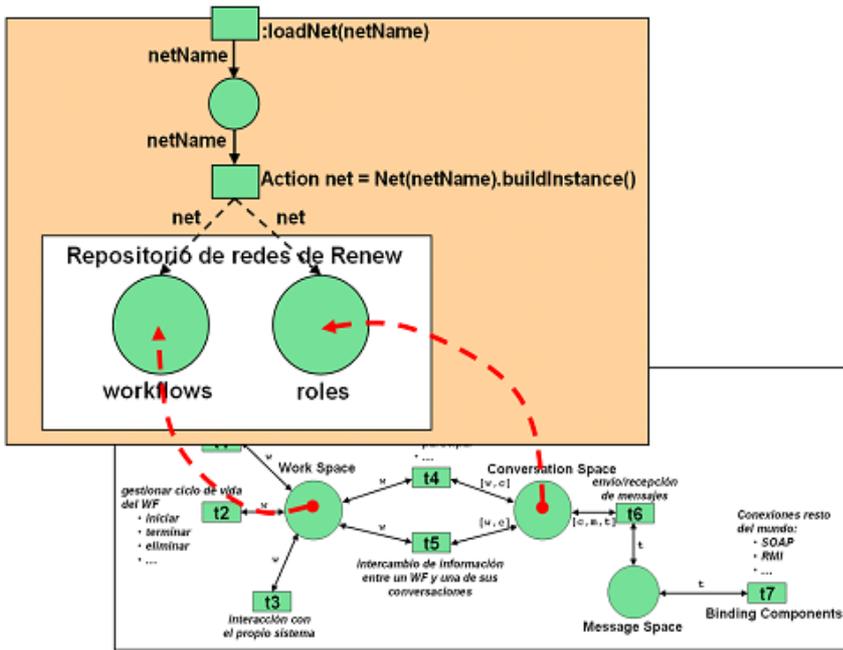


Figura 3. Componente para la carga dinámica de redes.

ción final y, por lo tanto, de la conexión de la plataforma con el mundo exterior.

5. Soporte para la configuración en tiempo de ejecución

Atendiendo a los requisitos de los procesos definidos en tiempo de ejecución, DENEb permite:

- Introducir nuevos *workflows* y protocolos de coordinación en tiempo de ejecución sin reiniciar la plataforma,
- Seleccionar desde un *workflow*, en función de determinados parámetros generados en tiempo de ejecución, el protocolo que se va a interpretar para conversar con otro proceso remoto,
- Recibir, como datos enviados de forma remota, *workflows* o roles de protocolos e iniciar su ejecución (más propio de entornos multiagente o de computación de procesos autónomos).

Esta funcionalidad es posible gracias a un componente de *carga dinámica de redes* y a un conjunto de *procesos de gestión* que se ejecutan sobre la propia plataforma.

5.1. Componente de carga dinámica de redes

En DENEb los *workflows* y los protocolos se representan como redes. Internamente, la herramienta Renew implementa las redes como clases Java, permitiendo crear nuevas instancias de una clase de red e iniciar su ejecución. No obstante, únicamente se pueden crear instancias de las clases de red que están "declaradas" en un repositorio interno de redes gestionado por Renew. Como se muestra en la figura 3 el componente de *carga dinámica de redes* ofrece operaciones

para introducir en tiempo de ejecución nuevas clases de redes en el repositorio de Renew. Estas clases pueden representar nuevos *workflows* o nuevos roles de un protocolo. Para facilitar la integración del componente en la plataforma de procesos y en la propia herramienta Renew, éste ha sido implementado también en base al paradigma de Redes-en-Redes.

Posteriormente, la plataforma puede crear instancias de los nuevos *workflows* publicados e iniciar su ejecución (más concretamente, desde la transición *t1*). Un *workflow* requiere cooperar con otros procesos para integrar su funcionalidad. La selección de los procesos con los que colabora y los protocolos utilizados para conversar con estos procesos puede realizarse en tiempo de diseño (ver figura 4-a), aunque una aproxima-

ción más dinámica debería permitir que también pudiera tener lugar en tiempo de ejecución. DENEb ofrece esta última posibilidad. En otras palabras, desde un *workflow* se puede seleccionar dinámicamente, en función de determinados parámetros generados en tiempo de ejecución, el rol de un protocolo publicado en el repositorio de redes de Renew e iniciar su ejecución para colaborar con el resto de procesos que intervienen en la conversación (figura 4-b). Evidentemente, esta estrategia dinámica de selección facilita que los procesos se acomoden de forma flexible a las condiciones cambiantes del entorno de ejecución. Sin embargo, exige el análisis previo de la compatibilidad sintáctica y semántica entre el *workflow* en ejecución y el nuevo rol de protocolo a interpretar.

5.2. Procesos de gestión

En general, los procesos de gestión implementan la funcionalidad de carácter horizontal integrada en la plataforma. Estos procesos conviven durante su ejecución con los procesos específicos de los distintos dominios de aplicación [4]. Un ejemplo interesante de proceso de gestión es el proceso responsable de recibir, como datos enviados de forma remota, nuevos *workflows* o roles de protocolos e iniciar su ejecución.

El funcionamiento de este proceso de gestión es muy simple. Durante su ejecución, el proceso monitoriza constantemente el repositorio de mensajes del *broker* integrado en la plataforma en busca de un tipo de tupla especialmente definido para encapsular *workflows* o roles de protocolos. Estas tuplas almacenan los *workflow* o los roles en formato PNML (*Petri Net Markup Language*), una especificación basada en XML para Redes de Petri. Cuando una de estas tuplas es insertada en el repositorio de mensajes, el proceso de gestión lee la tupla, transforma la representación PNML del *workflow* o del rol en una equivalente en forma de red y, utili-

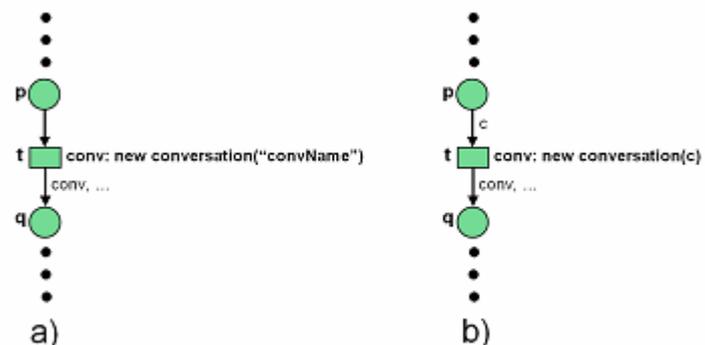


Figura 4. Diferentes estrategias para iniciar una conversación. a) El nombre de la conversación está "cableado" en el workflow. b) La conversación es seleccionada en tiempo de ejecución a partir de un conjunto (no predefinido) de posibles conversaciones

zando el componente de carga dinámica de redes, inserta la nueva red en el repositorio de redes interno de Renew. Una vez insertada, es posible crear un nuevo proceso o una nueva conversación como instancia de la nueva clase de red recibida.

6. Conclusiones

En este artículo se ha presentado una propuesta basada en el paradigma de Redes-en-Redes, como una alternativa válida a los actuales estándares y tecnologías de servicios Web, para la especificación, desarrollo y ejecución de procesos Web dinámicos.

A nivel de especificación, las Redes-en-Redes permiten abordar los requisitos dinámicos de los procesos Web y ofrecen suficiente capacidad expresiva para modelar *workflows* y protocolos de coordinación complejos. Además, el formalismo es directamente ejecutable. Por otro lado, la plataforma DENEb está basada en el mismo formalismo y, a diferencia de otras propuestas funcionalmente similares, no es un prototipo, ni un diseño arquitectural complejo pendiente de ser implementado, sino que gracias a la herramienta Renew es una implementación completa y operativa.

Agradecimientos

Los autores quieren agradecer a J.A. López

su colaboración en la implementación de la plataforma DENEb.

Este trabajo ha sido financiado por el proyecto TIN2006-13301 del Ministerio de Educación y Ciencia.

Referencias

- [1] B. Benatallah, F. Casati, F. Toumani. Web service conversation modelling. A cornerstone for e-business automation. *IEEE Internet Computing, January-February 2004*, págs. 46-54.
- [2] L. Ardissono, D. Cardinò, G. Petrone, M. Segnan. A framework for the server-side management of conversations with Web services. *Actas de la 13th International World Wide Web conference*, págs. 124-133, 2004.
- [3] B. Bjornstad, C. Pautasso, G. Alonso. Enforcing Web services business protocols at run-time: a process-driven approach. *International Journal of Web Engineering and Technologies, 2(4)*, págs. 396-407, 2006.
- [4] J. Fabra, P. Álvarez, P. Bañares, J.A. y Ezpeleta, J. A framework for the development and execution of horizontal protocols in open BPM systems, Conference on Business Process Management (BPM'06), pág. 209-224, Lecture Notes in Computer Science, Número 4102, Springer Verlag, 2006.
- [5] J. Fabra, P. Álvarez, J.A. Bañares, J. Ezpeleta. DENEb: Una plataforma para el desarrollo y ejecución de procesos Web dinámicos. *III Jornadas Científico-Técnicas en Servicios Web y SOA*

(JSWEB'07), pág. 11-18, ISBN 978-84-9732-604-9, 2007.

[6] R. Valk. Petri Nets as Token Objects - An Introduction to Elementary Object Nets. *19th Int. Conference on Application and Theory of Petri Nets (ICATPN'98)*, pág. 1-25, Lecture Notes in Computer Science, Número 1420, Springer Verlag, 1998.

[7] N. Carrero, D. Gelernter. Linda in context. *Communications of the ACM, 32(4)*, pág. 444-458, 1989.

[8] O. Kummer, F. Wienberg. Renew - the reference net workshop, Tool Demonstrations. *21st International Conference on Application and Theory of Petri Nets (ICATPN 2000)*, pág. 87-89, Computer Science Department, Aarhus University, Aarhus, Denmark, 2000.

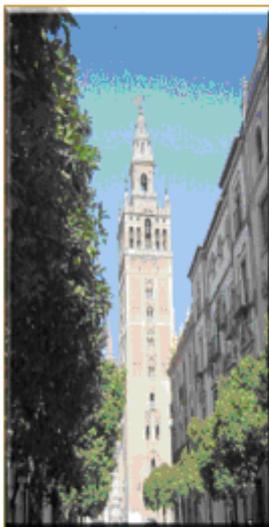
[9] J. Fabra, P. Álvarez, J. Ezpeleta. DR Linda: A Distributed Message Broker For Collaborative Interactions Among Business Processes. *Conference on Electronic Commerce and Web Technologies (EC-Web'07)*, pág. 212-221, Lecture Notes in Computer Science, Número 4655, Springer Verlag, 2007.

[10] R. Schmidt. Web services based architectures to support dynamic inter-organizational business processes. *International Conference on Web Services - Europe 2003 (ICWS'03)*, pág. 123-136, Lecture Notes in Computer Science, Número 2853, Springer Verlag, 2003.

[11] D. Moldt, S. Offermann, J. Ortmann. A Proposal for Petri Net Based Web Service Application Modeling. *4th. International Conference on Web Engineering (ICWE 2004)*, pág. 93-97, Lecture Notes in Computer Science, Número 3140, Springer Verlag, 2004.



IV Jornadas Científico-Técnicas en Servicios Web y SOA
Sevilla, 29-30 Octubre 2008



Objetivos

Estas Jornadas pretenden ser un punto de encuentro para profesionales, empresas e investigadores interesados en servicios Web y las arquitecturas orientadas a Servicios (SOA). En esta nueva edición, se prestará una especial atención (aunque no exclusiva) a trabajos de investigación y experiencias prácticas en las siguientes áreas:

1. Fundamentos de los servicios Web y SOA
2. Tecnología y servicios Web
3. Ingeniería de servicios Web
4. Aplicaciones de los servicios Web y SOA



Contacto:
jornadas2008@jsweb.es

Web:
www.jsweb.es